

Ejercicios Unidad 10

NOTA: Para probar estos ejercicios utilizar el archivo “Documentos.zip”.

Ejercicio 1. Implementa un programa que pida al usuario introducir por teclado una ruta del sistema de archivos (por ejemplo, “C:/Windows” o “Documentos”) y muestre información sobre dicha ruta (ver función más abajo). El proceso se repetirá una y otra vez hasta que el usuario introduzca una ruta vacía (tecla intro). Deberá manejar las posibles excepciones.

Necesitarás crear la función `void muestraInfoRuta(File ruta)` que dada una ruta de tipo `File` haga lo siguiente:

- Si es un archivo, mostrará por pantalla el nombre del archivo.
- Si es un directorio, mostrará por pantalla la lista de directorios y archivos que contiene (sus nombres). Deberá mostrar primero los directorios y luego los archivos.
- En cualquier caso, añada delante del nombre la etiqueta `[*]` o `[A]` para indicar si es un directorio o un archivo respectivamente.
- Si el path no existe lanzará un `FileNotFoundException`.

Ejercicio 2. Partiendo de una copia del programa anterior, modifica la función **`muestraInfoRuta`**:

- En el caso de un directorio, mostrará la lista de directorios y archivos en orden alfabético. Es decir, primero los directorios en orden alfabético y luego los archivos en orden alfabético. Te será útil `Arrays.sort()`.
- Añade un segundo argumento `‘boolean info’` que cuando sea `‘true’` mostrará, junto a la información de cada directorio o archivo, su tamaño en bytes y la fecha de la última modificación. Cuando `‘info’` sea `‘false’` mostrará la información como en el ejercicio anterior.

Ejercicio 3. Implementa un programa que haga lo siguiente:

- Cambiar el nombre de la carpeta ‘Documentos’ a ‘DOCS’, el de la carpeta ‘Fotografías’ a ‘FOTOS’ y el de la carpeta ‘Libros’ a ‘LECTURAS’
- Cambiar el nombre de todos los archivos de las carpetas FOTOS y LECTURAS quitándole la extensión. Por ejemplo, `‘astronauta.jpg’` pasará a llamarse `‘astronauta’`.

Ejercicio 4. Implementa un programa que cree, dentro de ‘Documentos’, dos nuevas carpetas: ‘Mis Cosas’ y ‘Alfabeto’. Mueve las carpetas ‘Fotografías’ y ‘Libros’ dentro de ‘Mis Cosas’. Luego crea dentro de ‘Alfabeto’ una carpeta por cada letra ‘A’, ‘B’, ‘C’.

Ejercicio 5. Escribe un programa que guarde en un fichero con nombre `primos.txt` los números primos que hay entre 1 y 500.

Ejercicio 6. Realiza un programa que lea el fichero creado en el ejercicio anterior y que muestre los números por pantalla.

Ejercicio 7. Implementa un programa que muestre por pantalla los valores máximos y mínimos del archivo 'numeros.txt'.

Ejercicio 8. Implementa un programa que pida al usuario un nombre de archivo A para lectura y otro nombre de archivo B para escritura. Leerá el contenido del archivo A (por ejemplo 'usa_personas.txt') y lo escribirá ordenado alfabéticamente en B (por ejemplo 'usa_personas_sorted.txt').

Ejercicio 9. El archivo 'alumnos_notas.txt' contiene una lista de 10 alumnos y las notas que han obtenido en cada asignatura. El número de asignaturas de cada alumno es variable. Implementa un programa que muestre por pantalla la nota media de cada alumno junto a su nombre y apellido, ordenado por nombre de mayor a menor.

Ejercicio 10. Implementa un programa que lea un documento de texto y muestre por pantalla algunos datos estadísticos: nº de líneas, nº de palabras, nº de caracteres y cuál es la palabra o palabras (si hay varias que se repitan el mismo número de veces) más común (y cuántas veces aparecen). Prueba el programa con los archivos de la carpeta 'Libros'. (Se recomienda usar un HashMap para la cuenta de las palabras)

Ejercicio 11. Crear una copia del archivo 'numeros.txt' e introducir en cualquier lugar del mismo varias líneas que no sean números. Repetir el Ejercicio 7, haciendo que funcione aunque el archivo contenga datos erróneos.

Ejercicio 12. Escribe un programa que guarde en un fichero el contenido de otros dos ficheros, de tal forma que en el fichero resultante aparezcan las líneas de los primeros dos ficheros mezcladas, es decir, la primera línea será del primer fichero, la segunda será del segundo fichero, la tercera será la siguiente del primer fichero, etc.

Hay que tener en cuenta que los ficheros de donde se van cogiendo las líneas pueden tener tamaños diferentes.

Ejercicio 13. Escribe un programa capaz de quitar los comentarios de un programa de Java.

Ejercicio 14. Escribe un programa que lea un archivo que contiene productos, cada producto tendrá nombre, cantidad y precio. Se deberá guardar en un objeto de tipo Producto, y mostrar todos los productos por pantalla, haciendo uso del método toString() de la clase.

Si de algún producto no es posible recuperar alguna de sus propiedades no se mostrará, aparecerá un error, pero el programa debe continuar mostrando el resto de productos.

Ejercicio 15. Escribe un programa que guarde productos en un fichero, las características de los mismos se solicitarán al usuario por consola. El programa nunca borrará productos que ya contenga el archivo, los añadirá a los existentes, y si el archivo no existe se creará. No está permitido leer el archivo en este programa, sólo escribirlo. El archivo generado tendrá un formato que permita el ejercicio 14 leerlo.

Ejercicio 16. Crear un programa que dada una ruta del sistema, muestre todo el contenido de la misma, y de sus directorios, hasta que no haya más subdirectorios, se recomienda utilizar recursividad.